

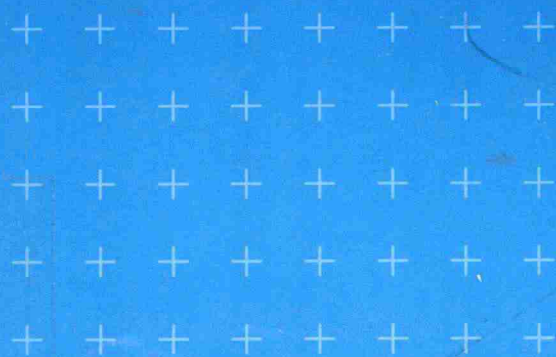
- 高等学校教材
- 北京市教育委员会科技发展计划项目资助
- 北京市优秀人才培养资助项目

Linux

程序设计

余有明 编著

—— 技术技巧与项目实践



附赠光盘

程序源代码

+

电子教案



化学工业出版社

- 高等学校教材
- 北京市教育委员会科技发展计划项目资助
- 北京市优秀人才培养资助项目

Linux 程序设计

——技术技巧与项目实践

余有明 编著



化学工业出版社

· 北 京 ·

本书以 Linux 技术为基础,以实战技巧为导向,以项目实践为演习,深入浅出地阐述了 Linux 程序设计的理论与实践。

全书共分十章,在内容安排上循序渐进,从了解 Linux 的基础知识到基本掌握 Linux 的编程环境的搭建,从在 Linux 环境下掌握基本的 Linux 技能到重点学习 shell 编程和 C 编程的理念与技术,在此基础上再深入学习文件系统编程、进程调度与通信编程、线程编程与网络编程技术,最后给出了巩固学习过程的实践项目,便于读者系统掌握 Linux 技术和学习编程的原理与实践。附带光盘收录了各章案例程序的源代码及电子教案。

本书结构合理、实例丰富,具有很强的启发性和实用性,可作为高等院校计算机、电气信息和控制类专业的本科生和研究生学习 Linux 系统与编程的专业基础教材,也可以供广大希望从事 Linux 环境下工作和转入嵌入式领域的科研与工程技术人员自学或参考使用。

图书在版编目(CIP)数据

Linux 程序设计——技术技巧与项目实践 / 余有明编著. —北京:化学工业出版社, 2012.6
高等学校教材
ISBN 978-7-122-14058-6

I. L… II. 余… III. Linux 操作系统-程序设计-高等学校-教材 IV. TP316.89

中国版本图书馆 CIP 数据核字(2012)第 072686 号

责任编辑:郎红旗 赵 鑫

文字编辑:张 赛

责任校对:陶燕华

装帧设计:史利平

出版发行:化学工业出版社(北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装:三河市延风印装厂

787mm×1092mm 1/16 印张 15¼ 字数 428 千字 2012 年 9 月北京第 1 版第 1 次印刷

购书咨询:010-64518888(传真:010-64519686)

售后服务:010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书,如有缺损质量问题,本社销售中心负责调换。

定 价: 33.60 元

版权所有 违者必究

前言

Preface



近年来, Linux 在操作系统和电子产品与“嵌入式世界”领域显示出新的活力和无限商机, 掌握和应用 Linux 系统有了更广阔的空间, 也是信息学科的莘莘学子高薪就业的基础与保障。

Linux 的魅力在于世界各地的计算机志愿者们为她贡献着自己的才华和智慧, 从初学者到专业人士, 从跨国公司的高管到经验丰富的黑客, 都在不断改进和维护着其系统的健康与繁衍生息。许多大学、研究机构、公司及家用 PC 机都在使用 Linux, 其年轻而富有朝气的生命力, 从诞生到现在, 乃至未来, 都具有显赫的地位和深远的影响力。

本书是基于 Linux 技术技巧与项目实践相结合的观点编著而成。介绍了 Linux 系统中的基础技术与 shell 编程和 C 语言编程的理论与实践, 在此基础上, 深入学习文件系统编程、进程调度与通信编程、线程与网络编程技术, 最后以实践项目为训练单元, 便于读者系统掌握 Linux 技术和学习编程的原理与实践, 相信读者在使用本书后会有所感触, 获益匪浅。

读者通过本书的学习能够快速地了解与学会 Linux 编程环境的搭建、Linux 应用技术和 shell 编程与 C 语言编程, 掌握其中的编程技术和技巧, 养成良好的编程习惯, 实现 Linux 环境编程技术的逐步提高。通过解析程序实例, 读者能够掌握 Linux 的关键操作与编程技术和技巧, 为培养开发大型应用程序的能力打下坚实的基础。

全书共分十章, 在内容安排上循序渐进。第 1 章 Linux 概述, 第 2 章构建 Linux 编程环境, 第 3 章 Linux 应用技术, 第 4 章 Shell 基础与编程技术, 第 5 章 C 程序设计技术, 第 6 章 Linux 文件系统与编程, 第 7 章进程调度与通信编程, 第 8 章线程编程技术, 第 9 章网络编程, 第 10 章 Linux 程序设计实践项目。基本主线是从基础到技能, 从编程理念到编程技术与技巧, 再到项目实践, 便于读者系统地学习和掌握 Linux 程序设计技术。

本书是编著者根据近年来的教学与项目开发经验并融合一些编程实例编著而成, 沿袭了 Linux 的成长特性, 还海纳百川般从网络中汲取了丰富的内涵——在此对这些素未谋面但确为本书贡献智慧的人们表示衷心的感谢。

本书由余有明博士拟定提纲并负责全书的编写工作, 吕红宇博士参与编写第 2、3 章, 赵国庆副教授参与编写第 4、5 章、向胜军老师参与编写第 6、7 章、张敏霞老师参与编写第 9、10 章。在此向他们表示衷心的感谢。

为了方便读者对教材中的程序进行验证和学习, 附带光盘收录了各章案例程序的源代码与电子教案等参考资料。建议读者不要直接使用源码, 而是尽量独立完成程序设计, 在调试出错或没有时间独立完成时再参考使用。

本书在编写过程中得到了“北京市教育委员会科技发展计划项目资助”(KM201010017007) 和“北京市优秀人才培养资助项目”的支持, 得到了华北电力大学的阎光伟博士、北京信息科技大学的刘旭红博士、北京工商大学的欧阳昱博士、北京印刷学院的冯柳平博士、北京石油化工学院的沙芸博士等有关专家教授和同行们的关心和帮助, 提出了很多宝贵的意见, 在此向他们表示衷心的感谢。

由于作者水平有限, 书中难免有不足和错误之处, 恳请各位专家和广大读者批评指正。

编著者
2012 年 5 月

目录

Contents



第 1 章	LINUX 概述	1
1.1	Linux 的渊源	1
1.2	Linux 的发行版本	2
1.3	Linux 的特性与生命力	2
1.4	Linux 用户应该掌握的 50 个常用命令	2
1.5	Linux 编程学习指导	2
1.6	实战技巧：Linux 平台下的软件自动安装与升级工具 APT 与 YUM	3
1.6.1	APT 的用法	3
1.6.2	YUM 的用法	4
第 2 章	构建 LINUX 编程环境	7
2.1	Linux 编程环境概述	7
2.1.1	编程环境的风格	7
2.1.2	Linux 下的编程环境与工具	7
2.1.3	Linux 下的编程分类	8
2.1.4	Linux 帮助应用指南	8
2.2	基本名词术语与硬件需求	9
2.2.1	系统操作术语	9
2.2.2	系统特性术语	10
2.2.3	系统硬件需求与安装准备	11
2.3	Linux 系统的多元化安装	11
2.3.1	裸机上的安装与卸载过程	11
2.3.2	多系统共享的安装过程	18
2.3.3	在 Windows 下 VMware 中的 Linux 的安装过程	19
2.3.4	用 Cygwin 软件创建 Linux 仿真环境	20
2.3.5	即时系统 Ubuntu Linux 和 WinPE	21
2.4	Linux 下软件的安装与卸载	22
2.4.1	二进制分发软件包的安装与卸载	23
2.4.2	软件包的安装与卸载	23
2.5	Xshell 的安装与使用	24
2.6	代码阅读工具 Source Insight	24
2.7	实战技巧：VMware tools 的安装与多平台下的文件夹共享	25

2.7.1	VMware tools 的安装	25
2.7.2	多平台下的文件夹共享	26

第3章 LINUX 应用技术 27

3.1	系统管理与维护技术	27
3.1.1	系统注销关机或重启命令	27
3.1.2	显示登录用户信息命令	28
3.1.3	账号管理命令	28
3.1.4	修改密码的 passwd 命令	30
3.1.5	显示与时间设置命令	30
3.1.6	磁盘空间管理命令	30
3.2	文件与目录操作技术	31
3.2.1	文件系统认识与权限设置	31
3.2.2	列表与分屏显示命令	32
3.2.3	切换创建与删除目录的命令	33
3.2.4	文件复制删除与合并命令	33
3.2.5	文件目录的移动与更名和路径显示命令	34
3.2.6	重定向与管道命令	35
3.2.7	改变文件访问权限与归属的命令	35
3.2.8	文件定位搜索与查找命令	36
3.3	前台程序的运行与状态切换	38
3.4	任务调度技术	38
3.4.1	系统工作调度	38
3.4.2	用户工作任务调度与删除	38
3.5	文件的打包与压缩解压缩	39
3.6	进程管理技术	39
3.6.1	命令 ps 的用法	39
3.6.2	命令 top 的用法	40
3.6.3	杀死进程命令 kill	41
3.7	控制提示符与命令组合	42
3.8	编辑器 Vi 的应用技术	42
3.8.1	Vi 的三种执行模式	42
3.8.2	基本操作	43
3.8.3	查找和替换	44
3.9	实战技巧：非空目录删除与终端字体大小改变	45
3.9.1	Linux 系统中非空目录的删除	45
3.9.2	改变 XTerm 的字体大小的方法	46

第4章 SHELL 基础与编程技术 47

4.1	Shell 概述	47
4.2	shell 程序设计基础	48
4.3	Shell 基本命令	50
4.3.1	内部命令	51

4.3.2	外部命令	51
4.3.3	命令分组	51
4.3.4	函数	51
4.3.5	Shell 的命令组合	53
4.3.6	Shell 特殊字符	54
4.4	Shell 变量	55
4.4.1	用户自定义变量	55
4.4.2	变量的类型声明	56
4.4.3	数组变量	56
4.4.4	变量的引用	57
4.4.5	键盘输入变量	58
4.4.6	位置变量	58
4.4.7	预定义特殊变量	59
4.4.8	环境变量与置换变量	59
4.4.9	算术运算	60
4.5	Shell 程序流程	60
4.5.1	测试语句 test	60
4.5.2	条件语句 if	62
4.5.3	开关语句 case	63
4.5.4	循环语句 for	64
4.5.5	循环语句 while	66
4.5.6	循环语句 until	67
4.5.7	循环控制命令 break 和 continue	67
4.5.8	流程控制 select	67
4.5.9	shift 语句和 repeat 语句	68
4.6	程序调试	69
4.7	Shell 编程综合范例	71
4.7.1	系统自动备份程序设计	71
4.7.2	终端颜色搭配的 color 脚本	71
4.7.3	目录菜单编制程序	72
4.7.4	问题交互图形化设计例程	72
4.7.5	计算两日期期间的天数例程	75
4.7.6	系统信息测试与显示例程	77
4.8	实战技巧：系统资源查看与多命令执行	79
4.8.1	使用 vmstat 命令来查看系统资源情况	79
4.8.2	使用分号同时执行多条命令	79

第 5 章 C 程序设计技术

5.1	管中窥豹学 Linux 下的 C 编程	80
5.2	GCC 编译器应用解析	81
5.2.1	编译器的工作过程	81
5.2.2	GCC 编译选项	81
5.2.3	GCC 编译过程与游戏编程	82

5.3	GDB 调试技术	85
5.3.1	调试编译与选项	86
5.3.2	程序调试设置	86
5.3.3	在线调试过程	87
5.3.4	在 GDB 中查看源程序	89
5.3.5	在 GDB 中查看栈信息	90
5.3.6	在 GDB 中查看运行数据	91
5.3.7	在 GDB 中改变程序的执行路线	93
5.3.8	在不同语言中使用 GDB	93
5.4	Make 工程管理器	94
5.4.1	Makefile 文件	94
5.4.2	make 命令	95
5.4.3	makefile 变量	96
5.4.4	Makefile 规则	98
5.5	多文件项目的 Autotools 应用	99
5.5.1	Autotools 基本工具	99
5.5.2	Autotools 制作 Makefile 基本流程例解	99
5.5.3	用 make 完成程序安装和发布	102
5.6	Linux 下的 C 程序设计实践	103
5.6.1	文件拷贝程序设计	103
5.6.2	猜数字游戏设计	104
5.6.3	进程调用程序设计	105
5.6.4	管道通信程序设计	106
5.6.5	“巧抓窃贼”程序设计	106
5.7	实战技巧: Linux 启动模式与定时任务设置	107
5.7.1	Linux 系统的命令行与 X 图形模式启动的设置与切换	107
5.7.2	定时任务设置	108
第 6 章	LINUX 文件系统与编程	109
6.1	文件系统概述	109
6.1.1	文件系统的目录树	109
6.1.2	文件系统的结构	110
6.1.3	Ext2 文件系统	111
6.1.4	VFS	115
6.2	设备文件	116
6.2.1	设备文件概述	116
6.2.2	设备驱动程序	116
6.2.3	中断处理	116
6.2.4	设备驱动程序的框架	117
6.3	文件的系统调用	117
6.3.1	文件的创建和读写	117
6.3.2	文件的各个属性	119
6.3.3	目录文件的操作	120

6.4	文件系统的设计与实现	121
6.5	实战技巧：消除 XWindows 下的死机现象	133
第 7 章	进程调度与通信编程	134
7.1	Linux 下的进程概述	134
7.1.1	进程的概念	134
7.1.2	Linux 进程的 PCB 结构	135
7.1.3	Linux 进程的组织方式	136
7.1.4	Linux 进程的异常检测	137
7.2	进程的系统调用	137
7.2.1	getpid 的用法	137
7.2.2	fork 的用法与多进程解惑	138
7.2.3	exit 和 _exit	139
7.3	进程间通信	141
7.3.1	管道通信	141
7.3.2	管道技术	142
7.3.3	无名管道编程	142
7.3.4	命名管道编程	145
7.3.5	信号通信	146
7.4	信号	148
7.4.1	信号常识	148
7.4.2	常用信号定义	148
7.4.3	信号的接收	148
7.4.4	信号的发送	150
7.4.5	共享内存	151
7.4.6	消息队列	152
7.5	守护进程	155
7.5.1	守护进程基础	155
7.5.2	守护进程的特点	156
7.5.3	守护进程的启动方式	156
7.5.4	守护进程的错误输出	156
7.5.5	守护进程的编程要点	157
7.5.6	守护进程案例	158
7.6	实战技巧：巧妙使用 Tab 键	160
第 8 章	线程编程技术	161
8.1	线程 thread 的内涵	161
8.2	线程调用	161
8.2.1	线程的定义	161
8.2.2	线程的分类	162
8.2.3	线程的实现	162
8.2.4	Linux 线程实现机制分析	163
8.3	多线程编程实践	167

8.4	线程的数据处理	169
8.4.1	线程数据	169
8.4.2	互斥锁	170
8.4.3	条件变量	173
8.4.4	信号量	175
8.5	多线程编程	177
8.5.1	线程的创建和终止	177
8.5.2	线程控制函数	177
8.5.3	线程间的互斥	178
8.5.4	线程间的同步	178
8.5.5	修改线程的属性	178
8.5.6	多线程编程解决生产者和消费者问题	180
8.6	实战技巧：拷贝与粘贴的快捷程序	182

第9章 网络编程 183

9.1	Socket 的概念	183
9.2	Socket 的建立与配置	183
9.3	Socket 的连接建立	185
9.4	数据传输	186
9.5	Socket 编程实例	187
9.5.1	服务器端的软件程序	187
9.5.2	客户端的软件程序	188
9.5.3	阻塞和非阻塞	190
9.5.4	基于 POP3 客户端实例	191
9.6	ping 命令解析	192
9.6.1	命令基本功能	192
9.6.2	命令 ping 源码例程	195
9.7	实战技巧：光驱与软驱的加载方法	197
9.7.1	光驱的使用	197
9.7.2	软驱的使用	197

第10章 LINUX 程序设计实践项目 199

10.1	实践项目一 虚拟机环境下的 Linux 安装	199
10.1.1	实践目的	199
10.1.2	实践准备	199
10.1.3	实践内容	200
10.2	实践项目二 Linux 的系统管理与基本命令	204
10.2.1	实践目的	204
10.2.2	实践内容	205
10.2.3	实践步骤	205
10.3	实践项目三 Vi 编辑器练习	211
10.3.1	实验目的	211
10.3.2	实验内容	211

10.3.3 实验步骤.....	211
10.4 实践项目四 Shell 程序设计.....	213
10.4.1 实践目的.....	213
10.4.2 实践内容.....	213
10.4.3 实践步骤.....	214
10.5 实践项目五 C 程序设计与多文件编译.....	216
10.5.1 实践目的.....	216
10.5.2 实践要求.....	216
10.5.3 实践内容.....	216
10.6 实践项目六 内存动态分区管理	219
10.6.1 实践目的.....	219
10.6.2 实践内容.....	219
10.6.3 实践程序及分析.....	219
10.7 实践项目七 Linux 文件操作编程	224
10.7.1 实践目的.....	224
10.7.2 实践内容.....	224
10.7.3 实践源程序清单.....	224
10.8 实践项目八 系统调用与设备管理	227
10.8.1 实践目的.....	227
10.8.2 实践内容.....	227
10.8.3 实践源程序清单.....	227
附录 LINUX 系统的安全防护技巧	229
参考文献.....	232

第 1 章

LINUX 概述



学习导航

- 1.1 Linux 的渊源
- 1.2 Linux 的发行版本
- 1.3 Linux 的特性与生命力
- 1.4 Linux 用户应该掌握的 50 个常用命令
- 1.5 Linux 编程学习指导
- 1.6 实战技巧: Linux 平台下的软件自动安装与升级工具 APT 与 YUM

1.1 Linux 的渊源

Linux操作系统的原始内核源于Minix的基本架构,由芬兰赫尔辛基大学的学生Linus Torvalds独立设计并最终完成,堪称编程史上的一次壮举,也是操作系统发展史上的奇闻轶事。1990年,Linus在读大学本科计算机专业时,在自己的机器上利用Tanenbaum教授设计的微型Unix操作系统Minix为平台,开发了属于他自己的第一个程序。第二年,他完成了令他誉满全球的操作系统Linux的最初内核。第三年,Linus把Linux奉献给自由软件基金会FSF(Free Software Foundation)的“GNU计划”,公布了全部源码,任何人都能从网上下载、分析、修改、添加新功能,甚至出售营利。不少专业人员认为Linux最安全、最稳定,对硬件系统最不敏感。短短几年的时间,Linux迅速成长,成为拥有庞大用户、性能稳定、功能完善的操作系统。目前的Linux装机总数已逾千万台,遍布全世界。在当今看重经济利益的商业社会,一个自由软件得到如此大众化的关心和关注,毫无疑问是IT业界的一个奇迹。现在发行的Linux操作系统软件包既包括完整的Linux操作系统,又包括文本编辑器、高级语言编译器等应用软件。还有带多个窗口管理器的X-Windows图形用户界面,用户如同使用微软的Windows一样,允许使用窗口、图标和菜单对系统进行GUI操作。

Linux 的开放性给我国的操作系统软件开发商提供了契机,便于我们开发具有自主知识产权的操作系统,打破国外厂商在计算机操作系统上的垄断。我国有多家软件公司致力于开发基于Linux 内核的操作系统平台,如中科红旗,并且有成功的产品应用在很多领域。

Linux在中国发展时间短,人才积累少,人才匮乏已成为产业发展的瓶颈。国内专业人才远远不如Windows系统下的雄厚,不论是数量还是质量都远远不及。Linux无限的应用前景和企业Linux人才的匮乏形成鲜明对比。据预测,国内未来5年Linux人才的需求将突破120万。

1.2 Linux 的发行版本

Linux的版本有多种不同的说法,但都大同小异。表1-1是Linux的主要发行版本。各版本的具体详情请参考其Internet站点。

表 1-1 Linux 的主要发行版本

版本名称	Internet 站点	基本特点
RedHat	http://www.redhat.com/	基于 RPM 包结构,安装简单,使用方便
Slackware	http://slackware.com/	基于 tgz 包结构,功能强,宜作服务器系统
Debian	http://www.debian.org/	基于 deb 包结构,收集的软件很全
OpenLinux	http://www.caldera.com/	基于 RPM 包结构,带有商业软件的演示版本
TurboLinux	http://www.turboLinux.com/	汉化较完整,有较强的配置管理功能
CosixLinux	http://www.cosixLinux.com.cn/	具有 Unix 韵味,但硬件驱动很陈旧
HappyLinux	http://www.happyLinux.com.cn/	联想 Linux 产品,汉化非常好,安装简单
RedFlag	http://www.redflag-Linux.com/	中科红旗软件产品,汉化完全,国内品牌

关于Linux的发音,世界各地不尽相同。Linux这个单词根据Linus Torvalds本人的发音应该是“哩呐克斯”,音标是[ˈli:nəks],重音在“哩”上。网络上有Linus Torvalds本人的一段录音,其内容是“Hello,this is Linus Torvalds and I pronounce Linux as Linux”。

Linux的获取主要通过网络下载或光盘方式获得。若从网络下载Linux,最好是到软件生产公司的官方网站下载比较可靠,参考表1-1并下载最新版本。也可以用有关关键词搜寻相关站点下载。当用户在学习和应用中遇到问题时,可以到Linux的Web网址和新闻讨论组寻找答案。如在<http://www.linuxjournal.com/>站点可以获得关于Linux杂志的最新消息,在各产品公司站点可以获得最新产品及功能版本,在<http://www.linux.org/>和<http://www.li.org/>等Linux国际机构站点上可以进行Linux在线讨论等。

1.3 Linux 的特性与生命力

Linux 操作系统在短短几年之内就得到了迅猛的发展,与其良好的特点是密不可分的。归纳起来,主要有这些基本也是很突出的特性:开放性、多用户、多任务、良好的用户界面、设备独立性、丰富的网络功能、可靠的系统安全与稳定性、良好的可移植性、软件安装的便利性、功能强大、对硬件要求很低、自由定制。

1.4 Linux 用户应该掌握的 50 个常用命令

- (1) 系统安装和登录命令(8个): login、shutdown、halt、reboot、install、mount、exit、last。
- (2) 文件处理命令(10个): file、mkdir、grep、dd、find、mv、ls、diff、cat、ln。
- (3) 系统管理相关命令(10个): df、top、free、quota、at、lp、adduser、groupadd、kill、crontab。
- (4) 网络操作命令(10个): ifconfig、ip、ping、netstat、telnet、ftp、route、rlogin、mail、nslookup。
- (5) 系统安全相关命令(7个): passwd、su、umask、chgrp、chmod、chattr、who。
- (6) 其他命令(5个): tar、unzip、gunzip、mttools、man。

1.5 Linux 编程学习指导

Linux 系统下的编程是令人兴奋的,操作者可以在 Linux 环境下编写出高质量的服务器程序

与应用程序，亲身感受到内核的强大动力。这里对如何学好 Linux 编程提出几点指导意见，以供读者参考。

(1) 明确学习目的。编程是 IT 业界人士的基本功，也是工作与生活的乐趣之一。优秀的程序员永远是被争夺的对象。学习编程能锻炼思维，享受创新快乐，走在高科技前沿。

(2) 打好基础。学习编程要具备一定基础，概括起来就是数学基础、逻辑思维能力和选择合适的入门语言。在 Linux 世界里，学习 C 语言是首选入门级语言，shell 编程也其乐无穷，效率很高，跨平台的可选择 Java 等。

(3) 注意理解重要概念。程序设计资料和教材里，能看到的是变量、函数、条件语句、循环语句等概念，但真正编程时，需要深入理解这些概念，把握语法与结构的应用实例，做到举一反三，触类旁通。

(4) 掌握编程思想。学习语言或开发工具，语法结构与功能调用是次要的，最主要的是算法与编程思想。例如，学习 C++ 就要学习系统的内在机理、线程；学习 COM 就要知道类、接口。先有思想才能游刃有余。

(5) 在做中学，多实践多交流。编程伊始要自己动手设计程序，不拘泥于固定思维方式，遇到问题要多想几种解决方案。多交流可了解别人的思维方式，不断吸收别人的长处，丰富编程实践，提高实践水平。

(6) 养成良好的编程习惯。编程入门容易提高难，坚持不懈是关键。编程风格好坏很大程度上会影响程序质量。良好的编程风格使程序结构清晰合理，代码便于维护。如代码缩进编排、变量命名规则的一致性、代码注释等。

(7) 上网学编程。网络无处不在，是很多编程大师们的良师益友。希望网络学习成为我们的必经之路。在网上可以学到迥异的编程思想、方法、经验和技巧。

(8) 加强计算机理论知识的再学习。学编程符合“理论→实践→再理论→再实践”的认识提高过程。程序人人可编，但发现编程能力很难再提高时，就要回头学习一些计算机科学和数学基础理论。学过之后，才能百尺竿头，更进一步。需要再学习的内容包括数据结构、算法理论、编译原理、操作系统原理、软件工程等计算机科学的理论基础和数理逻辑、代数系统、图论、离散数学等数学基础理论知识。

1.6 实战技巧：Linux 平台下的软件自动安装与升级工具 APT 与 YUM

在 Linux 平台下，用户最大的苦恼莫过于不会安装需要的程序，软件安装的便利性似乎落后于 Windows，但 APT 和 YUM 工具的出现给了用户全新的软件安装体验。作为软件自动安装最成熟的工具，用户只要告诉安装程序需要安装哪一款软件，安装程序就会自动下载并安装该软件的最新版本。当然，软件下载的前提是系统已经联网。如果要安装系统盘能够提供的软件包，可以用 `system-config-packages` 或 `redhat-config-packages`。下面分别介绍 APT 和 YUM 的用法。

1.6.1 APT 的用法

APT(Advanced Packaging Tool) 软件是一款轻量级的在线安装升级的软件包管理系统，`synaptic` 是图像化的 APT 软件包管理工具。

APT 主要有 `apt-cache`、`apt-cdrom`、`apt-config`、`apt-get` 四个命令。用户最常用的是 `apt-get`。这些命令的详细说明参见 `man` 手册，这里介绍其基本用法。

1. apt-get

(1) `apt-get update` ←更新本地 `apt-get` 的本地数据库，使其与服务器的 `pkglist` 文件同步。在

升级以前一般都要执行本命令实现与服务器的统一。

(2) `apt-get check` ←验证本地系统的完整性。

(3) `apt-get upgrade` ←类似于 `apt-get`，但是将安装所有的基础软件包，并尽力升级一切软件包，并在需要时安装新软件包。

(4) `apt-get install package_name` ←安装某个软件包及其相关软件包，如：`apt-get install openssh`，则既安装 `openssh`，同时也升级 `openssh`。

(5) `apt-get remove package_name` ←删除该软件包，同时会删除依赖于它的软件包

(6) `apt-get source package_name` ←下载软件包的 source RPM,例：

`#apt-get source --compile zhcon`

抓回 source RPM，编译为二进制的 RPM。--compile 参数功能等同于 `rpm -ba`。

(7) `apt-get clean` ←删除保存在缓存目录(/var/cache/apt/archives)下的下载包

(8) `apt-get upgrade package_name` ←升级指定包，并升级其依赖的软件包。

2. apt-cdrom

在安装系统后，可使用 `apt-cdrom` 把三张安装光盘加入到 APT 的 `sources.list` 中。先用“`#apt-cdrom add`”来加载第一张光盘，再运行两次“`#apt-cdrom add`”命令，加载另外 2 张安装光盘。在 `/etc/apt/sources.list` 中就会包含如下内容：

`#rpm cdrom:[Gaga V1.1 disc3]/ RedHat gaga`

`#rpm cdrom:[Gaga V1.1 disc2]/ RedHat gaga`

`#rpm cdrom:[Gaga V1.1 disc1]/ RedHat gaga`

这样以后需要时就可以直接使用 `apt-get` 安装以前未安装的软件包。

3. apt-cache

(1) `apt-cache showpkg` ←显示软件包的一些常规信息。

(2) `apt-cache stats` ←显示相关的统计信息。

(3) `apt-cache dump` ←显示缓存中的每个软件包的简要描述信息。

(4) `apt-cache unmet` ←显示不符合一致性的依赖关系。

(5) `apt-cache show` ←显示指定软件包的记录信息。类似于 `rpm -qi`。

(6) `apt-cache search` ←查找软件包，类似于 `rpm -qalgrep package_name`。

(7) `apt-cache depends` ←显示软件包的依赖性关系。

(8) `apt-cache pkgnames` ←列出所有的软件包。

4. apt-config

`apt-config dump` ←显示目前的配置信息。

1.6.2 YUM 的用法

YUM 是“Yellow dog Updater, Modified”的缩写，可以很方便地添加/删除/更新 RPM 包，配置文档简洁(`/etc/yum.conf`)，能同时配置多个资源库 Repository，能自动解决包的依赖性，便于管理大量系统的更新，保持和 RPM 数据库的一致性。不同版本的 Linux，在其中安装 YUM 的方法不同，请参考网络上的有关资料。

YUM 安装很简单，运行的命令行是：

`#rpm -ivh yum-*.noarch.rpm`

在第一次启用 YUM 之前需要导入系统的 RPM-GPG-KEY：

`#rpm --import /usr/share/doc/centos-release`

第一次使用 YUM 及其资源库更新时，YUM 会自动下载所有需要的 headers 放置于 `/var/cache/yum` 目录下，所需时间可能较长。

1. RPM 包的更新

`#yum check-update`

更新所有 RPM 包用命令：

```
#yum update
```

更新指定 kernel 和 kernel source 用：

```
#yum update kernel kernel-source
```

大规模版本升级：

```
#yum upgrade
```

2. RPM 包的安装和删除

如安装 xmms-mp3 包，使用：

```
#yum install xmms-mp3
```

删除倚赖性包，提示删除 licq-gnome,licq-qt,licq-text 时用：

```
#yum remove licq
```

3. YUM 缓存 (/var/cache/yum/) 的相关参数

清除缓存中 RPM 包文件使用命令：

```
#yum clean packages
```

清除缓存中 RPM 头文件使用命令：

```
#yum clean headers
```

清除缓存中旧的 RPM 头文件使用命令：

```
#yum clean oldheaders
```

4. 包列表

列出资源库中所有可以安装或更新的 RPM 包使用命令：

```
#yum list
```

列出资源库中特定可安装或更新及已安装包用：

```
#yum list Mozilla 或 #yum list mozilla*
```

列出资源库中所有可以更新的 RPM 包使用命令：

```
#yum list updates
```

列出已经安装的所有的 RPM 包使用命令：

```
#yum list installed
```

列出已经安装的但是不包含在资源库中的 RPM 包使用命令：

```
#yum list extras
```

5. RPM 包信息显示 (info 参数同 list)

列出资源库中所有可以安装或更新的 RPM 包的信息使用命令：

```
#yum info
```

列出资源库中特定的可以安装或更新以及已经安装的 RPM 包的信息使用命令：

```
#yum info mozilla 和 #yum info mozilla*
```

列出资源库中所有可以更新的 RPM 包的信息使用命令：

```
#yum info updates
```

列出已经安装的所有的 RPM 包的信息使用命令：

```
#yum info installed
```

列出已经安装的但是不包含在资源库中的 RPM 包的信息使用命令：

```
#yum info extras
```

6. 搜索 RPM 包

搜索匹配特定字符的 RPM 包使用命令：

```
#yum search mozilla
```

搜索有包含特定文件名的 RPM 包使用命令：

```
#yum provides realplay
```

7. 增加资源库

增加 rpm.livna.org 作为资源库使用命令，安装 Livna.org rpms GPG key:

```
#rpm --import http://rpm.livna.org/RPM-LIVNA-GPG-KEY
```

检查 GPG Key 使用命令:

```
# rpm -qa gpg-pubkey*
```

显示 Key 信息使用命令:

```
#rpm -qi gpg-pubkey-a109b1ec-3f6e28d5
```